EECS225d Audio Signal Processing in Humans and Machines **Spring 1995**

Lecture 34: 14 April 1995

Lecturer: Nelson Morgan *Scribe: Daishi Harada*

In this lecture, we discuss the use of feed-forward neural networks in the task of speech recognition. Often, actual systems are built using a mixture of neural networks and HMMs, and are hence called hybrid Connectionist/HMM ASR systems. Before doing so, we briefly review the previous lecture on HMMs.

34.1 Review of HMMs

The following assumptions are commonly made for hidden Markov models (HMMs):

First order Markov assumption The probability of the system being in a particular state at time t depends only on the state of the system at time $t-1$.

The output independence assumption The distribution over outputs of the system at time t depends only on the state of the system at time t.

Hence a first order HMM can generally be characterized by a transition probability matrix T and an emission probability vector E.

A HMM training problem, then, is essentially an estimation problem of the components of these two matrices. To solve this estimation problem, one typically uses either a form of EM called Baum-Welch, or an algorithm embedded with a Viterbi estimator of the maximum likelihood state sequence.

Recognition of an output sequence can be done using dynamic programming.

34.2 Overview of Artificial Neural Networks

We first note some myths/fallacies which are occasionally heard concerning ANNs:

- Humans use networks to hear; so should artificial systems.
- We should use ANNs because they can learn.
- Feature constraints not necessary with ANNs.

On the other hand, there are certain valid benefits to ANNs:

- Like HMMs, they can learn.
- Unlike most HMM approaches, learning can be discriminant.
- Can learn high order correlations between evidential sources without strong assumptions.

- Have a flexible architecture, e.g., for contextual inputs and feedback.
- Highly parallel and regular, good for hardware.

Of particular note is the way in which ANNs can be made to discriminate among classes. As noted in (Richard and Lippmann, 1991), ANN classifiers which have an output unit for each class, with the target vector having a 1 for the correct class and 0 elsewhere, and which use an error criterion such as mean squared error or relative entropy, can in fact be interpreted as learning the posterior probabilities of the class given the data.

34.3 ANNs in Speech Recognition

One common use of ANNs in speech recognition is as a preprocessor, for extracting likelihoods over phonemes from some representation of the audio signal (e.g., sequence of feature vectors). In order to capture the temporal structure of the signal, one can, for example, use a moving window over the input time series as the actual input to the network. One may generalize this to apply to the hidden layers also, so that each network layer can be construed as convolving one time series into another. Such a network architecture has been called a Time Delay Neural Network, and is described in e.g. (Waibel, *et al.* 1989).

34.4 Problems with ANNs, and what to do about them in real systems

Some of the problems which exist with attempting to use ANNs to solve the entire speech recognition problem are that:

- We don't know how to do training and recognition of connected speech units with ANNs only.
- We don't know how to explain an input sequence in terms of a sequence of output classes.

These are some of the reasons why many actual systems incorporate at least both ANNs and HMMs, taking advantage of their strengths and using one to compensate for the other's weaknesses. In particular, we may use

- ANNs for local discrimination, and
- HMMs for time integration and nonlinear time alignment.

Hence, the basic approach can be encapsulated as follows:

- Train a network for sound unit posteriors
- Generate HMM emission probabilities from the network.
- Align training using these probabilities.
- Retrain the network, stop when "done".
- Use the network for probabilities during recognition.

Clearly, a more precise notion of when training should be considered "done" would be desirable. One technique often used is cross validation on a test set distinct from the training set. The test set error gives us a estimate of the generalization error of the network on unseen data, and hence can be used as a stopping criterion.

The following parameter specifications are for systems currently being tested and used:

- 234 inputs (9 frames, 26 features).
- 1000 hidden units.
- 61 output units.
- 10^6 to 10^7 training frames.
- A few iterations.

References

M.D. Richard and R.P. Lippmann (1991), Neural Network Classifiers Estimate Bayesian a posteriori Probabilities, in *Neural Computation* pp. 461–483, Vol.3, No.4, Winter.

A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang (1989), Phoneme recognition using time-delay neural networks. in *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37:328–339, March.